



Kinematic Modeling and Observer Based Control of Soft Robot using Real-Time Finite Element Method

Zhongkai Zhang, Jeremie Dequidt, Alexandre Kruszewski, Frederick Largilliere, Christian Duriez

► To cite this version:

Zhongkai Zhang, Jeremie Dequidt, Alexandre Kruszewski, Frederick Largilliere, Christian Duriez. Kinematic Modeling and Observer Based Control of Soft Robot using Real-Time Finite Element Method. IROS2016 - IEEE/RSJ International Conference on Intelligent Robots and Systems, Oct 2016, Daejeon, South Korea. hal-01370347

HAL Id: hal-01370347

<https://inria.hal.science/hal-01370347>

Submitted on 22 Sep 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Kinematic Modeling and Observer Based Control of Soft Robot using Real-Time Finite Element Method

Zhongkai Zhang, Jeremie Dequidt, Alexandre Kruszewski, Frederick Largilliere and Christian Duriez

Abstract—This paper aims at providing a novel approach to modeling and controlling soft robots. Based on real-time Finite Element Method (FEM), we obtain a globally defined discrete-time kinematic model in the workspace of soft robots. From the kinematic equations, we deduce the soft-robot Jacobian matrix and discuss the conditions to avoid singular configurations. Then, we propose a novel observer based control methodology where the observer is built by Finite Element Model in this paper to deal with the control problem of soft robots. A closed-loop controller for position control of soft robot is designed based on the discrete-time model with feedback signal being extracted by means of visual servoing. Finally, experimental results on a parallel soft robot show the efficiency and performance of our proposed controller.

I. INTRODUCTION

Soft robots are defined as robotic systems of which compliance of underlying materials is similar to that of soft biological materials [1]. Due to their low stiffness of structure, soft robots can reduce the harm to interact with humans and show more adaptation to the environment. The advantages of soft robots make them suitable for medical applications, to manipulate objects with different shapes and to work in confined spaces. Compared with their rigid-bodies counterparts, soft robots have infinite degrees of freedom so that the modeling, control, and trajectory planning are more challenging. Traditional methods to model and control rigid robots are difficult to be implemented for soft robots and thus new approaches are required.

In the fields of soft robotics modeling, three main categories of kinematic models can be found: piece-wise constant curvature (PCC), non-constant curvature and Finite Element approximation. In the last ten years, researchers have done many works [2], [3] based on PCC model which can be considered as the simplest model of soft robot. However, the PCC model can not account for all kinds of soft robots, especially for the robots with non-constant sections. As a kind of non-constant curvature models, Cosserat geometrically exact models based on Cosserat rod theory has been explored [4], [5], [6]. The two kinds of models have been researched and used for model based control of soft robots. However, they are difficult to feature soft robots with relatively complex geometrical shapes or volume deformations. As a feasible numerical method, Finite Element Method (FEM) is usually limited to be used to provide a more realistic description of mechanics performance [7] due to its higher computational cost. The modeling of soft robots based on real-time FEM

was first proposed in [8] which opens a new research field for real time modeling and control of soft robots.

Based on the FEM model, open-loop control has been explored and experiments demonstrate its feasibility for control of soft robots with different sizes and shapes. [8] builds an inverse FEM model solved by iterative Gauss-Seidel algorithm to compute control input at each step. In [9], the inverse simulation method is applied to semi-automatic deformable registration for adaptive radiotherapy. In order to enable a high-rate control using the inverse FEM model, an asynchronous simulation framework is proposed based on multi-rate simulation [10]. Using a domain decomposition strategy, this control method is modified to control continuum robots with rigid vertebrae [11]. In the above control methods, no feedback signal from real soft robots is used to compute the control input. Therefore, these controllers have relatively lower accuracy and no robustness to disturbances. To avoid the difficulties of modeling, model-free control methods [12] were proposed to control soft robots using learning or optimization approaches. Generally speaking, model-free controllers have higher robustness to the change of environment but they are relatively complex with lower precision or longer time for learning.

To further the work in [8], the intention of this paper is to explore a novel method for model-based closed-loop control of soft robots. It is very difficult to get a relatively accurate model in analytical form for control design of soft robots. Real-time FEM provides a better choice to model the soft robots, especially for those with complex shapes or contact with the environment, with higher accuracy and acceptable computation time. Considering that the Jacobian matrix is an important index in the analysis and control of robots, we deduce the Jacobian matrix of soft robots based on a globally defined kinematic model. A feasible strategy to design closed-loop controller based on finite element model is to combine the feedback signal from the real robot and model information from the robot simulated by FEM. This generates a novel observer based control strategy where the observer is designed by FEM. There are two main methods in the sampled-data control literature [13]. In the first method, a continuous-time controller which is designed based on a continuous-time plant model is discretized for implication. While in the other method, a discrete-time model of the plant is deduced for design of a discrete-time controller. In this paper, we explore the second method using a new observer based control strategy.

Two main contributions are presented in our work. Based on real-time FEM, we get the kinematic equations which are globally defined to describe the movement of soft robots and then discuss the properties of soft-robot Jacobian Matrix. The second contribution is that we provide a novel method for closed-loop control of soft robot using a novel observer based control strategy to guarantee the stability and the minimal energy property of the robotic system. Besides, to our best knowledge, this is the first paper which proposes a closed-loop visual servoing controller based on Finite Element Method for the control of soft robot.

This article is structured as follows. In Section 2, a globally defined discrete-time kinematic model is obtained based on the real-time Finite Element Method. The features of Jacobian matrix for soft robots are also discussed in this section. The principle of the proposed observer based control and its design based on the discrete-time model are shown in Section 3. Section 4 shows the experimental results on a parallel soft robot using the controller designed in Section 3. Then conclusions and future researches are presented in Section 5.

II. SOFT-ROBOT JACOBIAN MATRIX BASED ON FINITE ELEMENT METHOD

A. Kinematic Modeling

This section is partly built on the modeling method in [8], [14], [15] which propose to build the equilibrium function for soft objects in the constraint space. The models proposed by these papers describe the evolution of the steady state (position of each nodes of the FEM) of the system when a given input variation is applied. The are expressed in the form of implicit discrete-time equations.

At each sampling time T (T is the sampling interval time which is very small and is equal to the computation time of the relative step), the quasi-static equilibrium function (i.e for low velocities) of the entire robot is given by ¹:

$$\mathbf{f}_{ext,k} - \mathbf{f}(\mathbf{x}_k) + \mathbf{H}_k^T \lambda_k = 0 \quad (1)$$

where \mathbf{x}_k is the state of the robot (position of each node at sample time k). \mathbf{f}_{ext} is the external loads (like the gravity) and $\mathbf{f}(\mathbf{x})$ is the internal stiffness forces at a given position \mathbf{x} of the FEM nodes. $\mathbf{H}^T \lambda$ represents the contributions of the actuators and the contact forces (where \mathbf{H}^T provides the directions of the forces on the nodes imposed by each actuator and contact force, and λ is their unknown contributions).

A linearization of the internal forces is computed at the $(k+1)th$ sampling time using the FEM model:

$$\mathbf{f}(\mathbf{x}_{k+1}) \approx \mathbf{f}(\mathbf{x}_k) + \mathbf{K}(\mathbf{x}_k) d\mathbf{x}_{k+1} \quad (2)$$

where $\mathbf{K}(\mathbf{x}_k)$ is the tangent stiffness matrix that depends on the actual positions of the nodes and $d\mathbf{x}_{k+1}$ is the displacement between consecutive positions ($d\mathbf{x}_{k+1} = \mathbf{x}_{k+1} - \mathbf{x}_k$). Substituting Eq. (2) into Eq. (1), the equilibrium equation at each step can be established:

$$\mathbf{K}(\mathbf{x}_k) d\mathbf{x}_{k+1} = \mathbf{f}_{ext,k+1} - \mathbf{f}(\mathbf{x}_k) + \mathbf{H}_{k+1}^T \lambda_{k+1} \quad (3)$$

In motion space, the matrix \mathbf{K} is often very large in dimension so that the direct computation of equilibrium equation is expensive. Instead, using the Schur complement of the constraint problem, the FEM model equation can be projected into the constraint space that drastically reduces its size. Then, we have:

$$\delta_{e,k} = \underbrace{[\mathbf{H}_e(\mathbf{x}_{k-1}) \mathbf{K}^{-1}(\mathbf{x}_{k-1}) \mathbf{H}_a^T(\mathbf{x}_{k-1})]}_{\mathbf{W}_{ea}(\mathbf{x}_{k-1})} \lambda_{a,k} + \delta_{e,k}^{free} \quad (4)$$

$$\delta_{a,k} = \underbrace{[\mathbf{H}_a(\mathbf{x}_{k-1}) \mathbf{K}^{-1}(\mathbf{x}_{k-1}) \mathbf{H}_a^T(\mathbf{x}_{k-1})]}_{\mathbf{W}_{aa}(\mathbf{x}_{k-1})} \lambda_{a,k} + \delta_{a,k}^{free} \quad (5)$$

where δ_e and δ_a are, respectively, positions of the end-effector and the actuator in constraint space. While δ_e^{free} and δ_a^{free} are positions given at the free configuration ($\lambda_a = 0$). \mathbf{W}_{ea} and \mathbf{W}_{aa} are the basic matrices and are homogeneous to a compliance. Using \mathbf{W}_{ea} , we can get a measure of the mechanical coupling between effector and actuator, and with \mathbf{W}_{aa} , the coupling between actuators. The matrix \mathbf{K} can be factorized using a *LDL* decomposition, even for a large number of nodes and actuators. In order to obtain the mechanical coupling between effectors and actuators, the Jacobian matrix \mathbf{H}_e is defined as an additional constraint at the tip of the effector.

At the $(k+1)th$ step, Eq.(4) can be written as:

$$\delta_{e,k+1} = \mathbf{W}_{ea}(\mathbf{x}_k) \lambda_{a,k+1} + \delta_{e,k+1}^{free} \quad (6)$$

Combining Eq. (4) and Eq. (6), and considering the reasonable assumption of small displacements, i.e. $\mathbf{W}_{ea}(\mathbf{x}_k) = \mathbf{W}_{ea}(\mathbf{x}_{k-1})$ and $\delta_{e,k}^{free} = \delta_{e,k+1}^{free}$, we obtain the discrete-time model of soft robots:

$$\delta_{e,k+1} = \delta_{e,k} + \mathbf{W}_{ea}(\mathbf{x}_k) \Delta \lambda_{a,k} \quad (7)$$

where $\Delta \lambda_{a,k} = \lambda_{a,k+1} - \lambda_{a,k}$.

Similarly, from Eq. (5), we have

$$\delta_{a,k+1} = \delta_{a,k} + \mathbf{W}_{aa}(\mathbf{x}_k) \Delta \lambda_{a,k} \quad (8)$$

The compliance matrices $\mathbf{W}_{ea}(\mathbf{x}_k)$ and $\mathbf{W}_{aa}(\mathbf{x}_k)$ can not be precomputed because their values change at each iteration. Based on the fact that the soft materials have slow dynamics and are not subject to high-frequency deformation, we can make an assumption that the matrices $\mathbf{W}_{ea}(\mathbf{x}_k)$ and $\mathbf{W}_{aa}(\mathbf{x}_k)$ are constant between two sample times.

At the next step, the matrices $\mathbf{W}_{ea}(\mathbf{x}_k)$ and $\mathbf{W}_{aa}(\mathbf{x}_k)$ are computed under the configuration of the robot which is updated by

$$\mathbf{x}_{k+1} = \mathbf{x}_{free,k+1} + \mathbf{K}_k^{-1} \mathbf{J}_k^T \lambda_k \quad (9)$$

¹This function is built on the assumption of low velocities, so the dynamic feature (inertia force) of the robot is ignored.

Remark 1. $\mathbf{W}_{ea}(\mathbf{x}_k)$ and $\mathbf{W}_{aa}(\mathbf{x}_k)$ are not time-varying matrices. Instead, their values depend on the positions of all nodes (the configuration of soft robot) in the Finite Element Model. In this paper, $\mathbf{W}_{ea}(\mathbf{x}_k)$ and $\mathbf{W}_{aa}(\mathbf{x}_k)$ mean the values of \mathbf{W}_{ea} and \mathbf{W}_{aa} at the k th sampling time.

B. Jacobian Matrix

In the research field of robotics, the Jacobian matrix (representing the differential relationship between the actuator displacement and the effector motion) is heavily used to analyze and control robots. For rigid robots, assuming that the relationship between the task space and joint space is $\mathbf{x} = f(\mathbf{q})$, the kinematics model of the robot can be written as:

$$d\mathbf{x} = \mathbf{J}d\mathbf{q} \quad (10)$$

where \mathbf{J} is the Jacobian matrix and can be expressed as:

$$\mathbf{J} = \frac{df(\mathbf{q})}{d\mathbf{q}} \quad (11)$$

However Eq. (10) and Eq. (11) can not be written using the geometry of soft robots, as it is done traditionally in rigid robots. It needs to account for the mechanical coupling between actuators and between actuators and effectors created by the deformation [8]. In the control of soft robotic systems, the position of end-effector δ_e is the output variable in task space and δ_a is the input signal to the actuator in joint space. Combining kinematic equations Eq. 7 and Eq. 8, the differential relationship for soft robots can be written as:

$$d\delta_e = \mathbf{W}_{ea}\mathbf{W}_{aa}^+ \cdot d\delta_a \quad (12)$$

where $d\delta_e = \delta_{e,k+1} - \delta_{e,k}$ and $d\delta_a = \delta_{a,k+1} - \delta_{a,k}$. The Jacobian matrix for soft robots is $\mathbf{J} = \mathbf{W}_{ea}\mathbf{W}_{aa}^+$. For the case of 3D movement, there exist singular configurations when $\text{rank}(\mathbf{J}) < 3$ which should be avoided for control design and analysis of the workspace. Note that Eq. (12) describes the 3D position movement of soft robots, the vector δ_e always contains three rows and thus we should make sure that

$$\text{rank}(\mathbf{W}_{ea}\mathbf{W}_{aa}^+) = 3 \quad (13)$$

The matrix \mathbf{W}_{aa} is positive-definite if there is no linear dependency between lines of \mathbf{H}_a , which can be realized by placing the actuators on different nodes of the FEM with different directions. Considering this invertible property of the matrix \mathbf{W}_{aa} , we have

$$\text{rank}(\mathbf{W}_{ea}) = \text{rank}(\mathbf{W}_{ea}\mathbf{W}_{aa}^+) = 3 \quad (14)$$

Assuming that there are n actuators, \mathbf{W}_{aa} is a $n \times n$ square matrix and \mathbf{W}_{ea} has 3 rows and n columns. Some conclusions about soft robots, based on Eq. (13) and Eq. (14), can be made as follows:

If $n < 3$, the robot is a less freedom system which means that the 3D movement is limited no matter what kind of controller is applied.

If $n = 3$, from Eq. (13), the matrix \mathbf{W}_{ea} should be invertible. Then the unique solution of control input is obtained.

If $n > 3$, the matrix \mathbf{W}_{ea} should be a column non-singular matrix ($\text{rank}(\mathbf{W}_{ea}) = 3$) and thus the pseudoinverse can be computed by $\mathbf{W}_{ea}^+ = \mathbf{W}_{ea}^T (\mathbf{W}_{ea}\mathbf{W}_{ea}^T)^{-1}$ which can be used for control allocation.

Therefore, in the design and control of soft robots, we should make sure that the matrix \mathbf{W}_{ea} is a column non-singular matrix, the matrix \mathbf{W}_{aa} is invertible, and the number of actuators is larger than 2. Then the position of soft-robot end-effector can be controlled freely in 3D workspace.

III. CONTROLLER DESIGN

A. Observer based Control

The objective of this section is to design a control law so that the end-effector of the robot can reach its desired position $\mathbf{y}_{e,d}$ asymptotically. The controller is designed based on the kinematic equation Eq. (7) which can be rewritten as the state-space model:

$$\delta_e(k+1) = \delta_e(k) + \mathbf{B}(k)\mathbf{u}(k) \quad (15)$$

$$\mathbf{y}(k+1) = \delta_e(k+1) \quad (16)$$

where δ_e is the state variable of the system (i.e. the position of the end-effector δ_e). $\mathbf{B}(k)$ is the input matrix and is equal to $\mathbf{W}_{ea}(\mathbf{x}_k)$. The tension input is $\mathbf{u} = \Delta\lambda_a$, and the output of the system is $\mathbf{y} = \delta_e$. Based on the discrete-time model Eq. (15), a feasible controller can be designed as

$$\mathbf{u}(k) = -\mu\mathbf{B}^+(k)(\mathbf{y}_{e,k} - \mathbf{y}_{e,d}) \quad (17)$$

which is developed in the next subsection.

In traditional control implementation, control input is computed by control law which is only based on the information in the real space². This is a feasible way to design controllers whose feedback information can be measured or observed in application (usually it is for systems with a few degree of freedoms). However, it is not a good idea for complex systems (like soft robots accurately modeled by FEM and controlled by controller $-\mu\mathbf{B}^+(k)(\mathbf{y}_{e,k} - \mathbf{y}_{e,d})$) because neither $\mathbf{W}_{ea}(\mathbf{x}_k)$ nor $\mathbf{W}_{aa}(\mathbf{x}_k)$ can be computed from the information of real robots. In this subsection, we introduce a novel implementation of observer based control for complex systems. The principle of this control strategy is shown in Fig. (1).

The simulation model serves as an observer to obtain the values of matrices $\mathbf{W}_{ea}(\mathbf{x}_k)$ and $\mathbf{W}_{aa}(\mathbf{x}_k)$ at each sampling time. The tension $\Delta\lambda_a$ is computed using information from two basic components: one is the observer built by the Finite Element Model (to obtain matrix \mathbf{W}_{ea}) and the other is the robotic system (to obtain the output δ_e). The input signal δ_a which is computed by substituting $\Delta\lambda_a$ and \mathbf{W}_{aa} (obtained from Finite Element Model) into Eq. (8) is used to drive both the simulation model and the real robot. At the next step computation, the model of the observer is updated based on Eq. (9) to get the new updated matrices \mathbf{W}_{ea} and \mathbf{W}_{aa} .

²If the feedback information for the controller can be measured, different sensors can be applied to extract this information. For information which is not accessible directly, observers are used for real application.

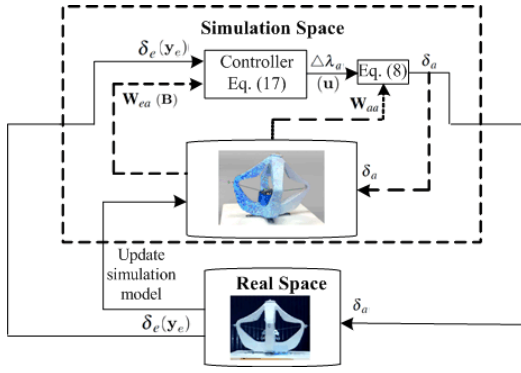


Fig. 1. Implementation of controller proposed in this paper. Information from both the real space and the simulation space is needed to compute the control input. The information from robot in real space is used to generate feedback signal for the controller and to adjust the simulation model step by step. The model in simulation space can provide other information (like the value of matrix \mathbf{B} in Eq. (15)) which is needed for the controller.

Considering that it is impossible to take every disturbance into consideration, the model in simulation space is usually an approximation of the robot. Therefore, if the differences between the accurate model and the simulation one are obvious, the approximated model can be adjusted using the information from real space to provide an accurate observer.

If the number of actuators is equal to the dimension of end-effector space, we have $\mathbf{W}_{ea,k}^+ = \mathbf{W}_{ea,k}^{-1}$ and get the unique solution of control input. Otherwise, the least 2-norm solution of the control input can be obtained using the control law Eq. (18). From the definition of 2-norm solution, we have $\|\lambda_a\|_2 = (\lambda_a^T \lambda_a)^{\frac{1}{2}} = (\sum_{i=1}^n \lambda_{ai}^2)^{\frac{1}{2}}$. The unique solution of displacement of actuators is obtained by computing Eq. (8) at each step. Therefore, the displacement of actuators also meets the condition: $\|\Delta\delta_a\|_2 = (\Delta\delta_a^T \Delta\delta_a)^{\frac{1}{2}} = (\sum_{i=1}^n \Delta\delta_{ai}^2)^{\frac{1}{2}}$. Furthermore, we have $\|\lambda_a\|_2 \cdot \|\Delta\delta_a\|_2 = \lambda_a^T \Delta\delta_a = E^3$ which means the minimal control effort is applied to deform the robot at each step.

Unlike the methods to compute the control input at each step from Eq. (4) using GS [8] or QP [16] algorithms, controllers in this paper are designed based on Lyapunov method and are implemented using observer based control strategy. Three assumptions are made to simplify the design process.

1. The finite element model of soft robots in simulation space is a precise model of the real robot and constitutes a perfect state observer;
2. The robot workspace contains no singular configuration;
3. The actuators are not constrained (no saturation).

B. Control based on Discrete-time Model

Based on Lyapunov method, the position control law $\Delta\lambda_a$ for soft robot is designed in this subsection and is shown in Result 1.

³The deformation energy E in each step is linked to the mechanical work of the forces exerted by the actuators: $E = \lambda_a^T \Delta\delta_a$.

Result 1. Considering the closed-loop system, described by Eq. (7), the feedback control law to compute the forces is designed as

$$\mathbf{u}(k) = -\mu \mathbf{B}^+(k) (\mathbf{y}_{e,k} - \mathbf{y}_{e,d}) \quad (18)$$

Then for any μ s.t. $0 < \mu < 2$, the system is globally asymptotically stable.

Proof: The error signal \mathbf{e} is defined as: $\mathbf{e} = \mathbf{y} - \mathbf{y}_{e,d}$, then we consider the Lyapunov candidate function:

$$V = \mathbf{e}^T \mathbf{e} \quad (19)$$

The variation of Lyapunov function can be written as:

$$\begin{aligned} \Delta V &\triangleq V(k+1) - V(k) \\ &= \mathbf{e}_{k+1}^T \mathbf{e}_{k+1} - \mathbf{e}_k^T \mathbf{e}_k \\ &= (\mathbf{y}(k+1) - \mathbf{y}_{e,d})^T (\mathbf{y}(k+1) - \mathbf{y}_{e,d}) - \\ &\quad (\mathbf{y}(k) - \mathbf{y}_{e,d})^T (\mathbf{y}(k) - \mathbf{y}_{e,d}) \\ &= \mathbf{y}^T(k+1) \mathbf{y}(k+1) - \mathbf{y}^T(k) \mathbf{y}(k) - 2\mathbf{y}^T(k+1) \mathbf{y}_{e,d} \\ &\quad + 2\mathbf{y}^T(k) \mathbf{y}_{e,d} \end{aligned} \quad (20)$$

Substituting Eq. (7) into Eq. (20), we have

$$\begin{aligned} \Delta V &= (2\mathbf{y}^T(k) + (\mathbf{B}(k) \mathbf{u}(k))^T - 2\mathbf{y}_{e,d}^T) \\ &\quad \cdot (\mathbf{B}(k) \mathbf{u}(k)) \end{aligned} \quad (21)$$

Substituting Eq. (18) into Eq. (21), the final form of ΔV is

$$\Delta V = -\mu(2 - \mu) \mathbf{e}^T(k) \mathbf{e}(k) \leq 0 \quad (22)$$

In Eq. (22), $\Delta V < 0$ stands for all $\mathbf{e} \neq 0$, so that all solutions converge to the set $S := \{\mathbf{y} \mid \mathbf{y}(k) = \mathbf{y}_{e,d}\}$. From Eq. (18), we can further conclude that the input μ converges to zero, so that Eq. (8) is also stable. ■

Then the incremental displacement of actuators is obtained by substituting $\mathbf{u}(k)$ into Eq. (8):

$$\Delta\delta_a(k) = -\mu \mathbf{J}(k)^+ (\mathbf{y}_{e,k} - \mathbf{y}_{e,d}) \quad (23)$$

IV. EXPERIMENTS

A parallel soft robot with four legs (see Fig. 2) is considered to show the performance of controller Eq. (18). The soft robot is made of silicone with a Young Modulus of $150kPa$ and is driven by four inelastic cables (pulled by servomotors with displacement controller) at each leg. The position of end-effector is chosen as the feedback signal and is measured at each step by the motion capture system (*OptiTrack* with the accuracy of $0.1mm$), which constitutes a position-based visual servoing system. The soft robot is modeled in SOFA framework where the value of matrices \mathbf{W}_{ea} and \mathbf{W}_{aa} is computed at each step based on FEM. Instead, vectors δ_e and $\delta_{e,d}$ are measured by the cameras in the real space.

There are two feasible ways to implement the control of soft robots. When the forces of the cables are computed by Eq. (18), torque control can be applied for the control of

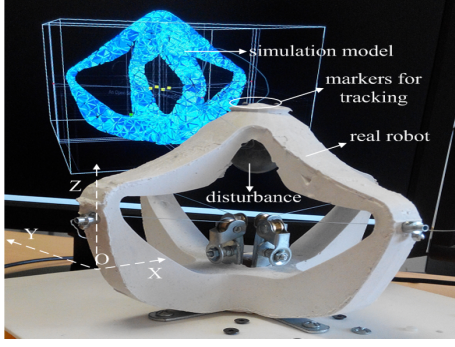


Fig. 2. Picture of soft robot and its simulation model. The parallel soft robot is made of silicone and is actuated with cables placed on the middle of each leg. A rigid ball is used to generate the disturbance on the robot in order to test the performance of the controller. A simulation model is built in the scene of SOFA.

servomotors. In the second method which is applied by this paper, we get the displacement of the cables by substituting Eq. (18) into Eq. (8) at first and then control the servomotors by the position control method. The other implementation of the second method is to compute the displacement of cables directly using control law $\Delta \delta_{a,k} = -\mu \mathbf{W}_{aa} \mathbf{B}^+(k) (\mathbf{y}_{e,k} - \mathbf{y}_{e,d})$ which is obtained by the model $\delta_{e,k+1} - \delta_{e,k} = \mathbf{W}_{ea} \mathbf{W}_{aa}^{-1} \Delta \delta_{a,k}$ (combining Eq. (7) and Eq. (8)). In this method, the controller is deduced using the same method for Eq. (18) and assuming that the matrix \mathbf{W}_{aa} is invertible. In the second method, when Eq. (18) is substituting into Eq. (8), we can also obtain the control law which is the same with that by the third method.

Positioning and tracking performances of both closed-loop controllers and open-loop controllers⁴ are shown in this section. The position errors of the end-effector between its real position and desired position at every step are depicted in Fig. (3), Fig. (4), and Fig. (5), where the red lines depict the changes of position errors using closed-loop controllers while the green lines show that using open-loop controllers. Control parameters and exact average errors are shown in Tab. (I) where the shorthand notations of control methods in experiments are: closed-loop positioning control (CPC), open-loop positioning control (OPC), closed-loop positioning control with disturbance (CPCD), open-loop positioning control with disturbance (OPCD), closed-loop tracking control (CTC), and open-loop tracking control (OTC).

A. Positioning

In positioning experiments, we consider the performances of both the closed-loop controller and the open-loop controller in two situations: with and without disturbance. The unknown disturbance is generated by an unmodeled rigid ball swaying on the end-effector of the soft robot (see Fig. 2). The initial tensions of four cables are set as 0.58 N which drives the end-effector to the initial position $[3, -122, 129]$ (mm) for the robot without disturbance and $[4, -121, 111]$ (mm) with disturbance. The desired position is $[10, -110, 160]$ (mm) for each case.

⁴In open-loop controller, the same control law (18) is used with the output feedback signal being obtained from the simulation model.

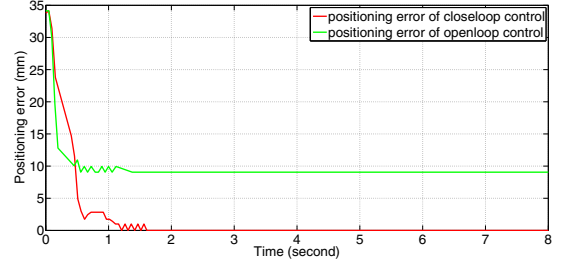


Fig. 3. Comparison of positioning errors between open-loop control and closed-loop control

The Fig. (3) shows the performances of open-loop control and closed-loop control without the disturbance in the environment. The positioning error is zero after 1.6 seconds using the closed-loop control, while in open-loop control, due to the modeling error of the simulation model, the position error does not reach zero. As shown in Tab. (I), the average positioning error within 8 seconds using closed-loop control is about 9.74 mm lesser than that using open-loop control.



Fig. 4. Comparison of positioning errors between open-loop control and closed-loop control with unknown disturbance

The Fig. (4) presents the positioning errors in the appearance of disturbance. The closed-loop control is effective for positioning task with the positioning error being zero after 3.2 seconds and the average error being 3.23 mm (see Tab. (I)). However, for open-loop control, the positioning error is larger than that without disturbance.

B. Tracking

The tracking performances are presented in this subsection with the initial tension of cables being 0.87 N for an initial position of the end-effector at $[4, -121, 141]$ mm. The trajectory of the tracked object is described by $[2 + 20 \sin(t/5), -122 + 20 \cos(t/5), 150]$ (mm). Another situation where the moving object has no preassigned path is also considered.

As depicted in Fig.(5), the tracking performance is much better with closed-loop control than that with open-loop control (see average errors from Tab (I) : 1.94 mm for CTC and 13.13 mm for OTC). Because of the loss of dynamic information, the vibration of the soft robot can not be suppressed and the tracking error exists for both control methods no matter how long the tracking time is.

As shown in Fig. (6), the robot can track a moving object without a preassigned path using the controller proposed in

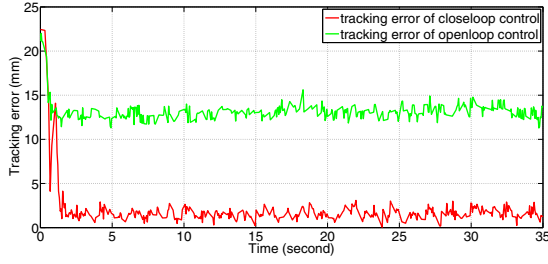


Fig. 5. Comparison of tracking errors between open-loop control and closed-loop control

TABLE I
CONTROL PARAMETERS AND AVERAGE ERRORS IN EXPERIMENTS

Experiments	Control parameters	Average errors (mm)
CPC	0.12	2.46
OPC	0.5	12.20
CPCD	0.06	3.23
OPCD	0.5	15.84
CTC	0.12	1.94
OTC	0.5	13.13

this paper. The positions of both the object and the end-effector are extracted by cameras at each step. At the initial time, the position of the object is converted to coincide with the position of the end-effector by the translational motion ($T_{eo} = T_o - T_e$ shown in Fig. (6)). Then for the following steps, what the end-effector tracks is the equivalent object of which position is obtained by the same translational motion as that in the initial time.

V. CONCLUSION

This paper shows a novel method for kinematic modeling and control of soft robots. The discrete-time model is deduced based on real-time Finite Element Method and then a novel observer based control method (the observer is built by Finite Element Model) is proposed to control soft robots with complex structure. Two kinds of experiments (positioning and tracking) are implemented and show the effectiveness of the proposed method.

The application of this approach to different soft robots will be explored in the future work. The performance of our

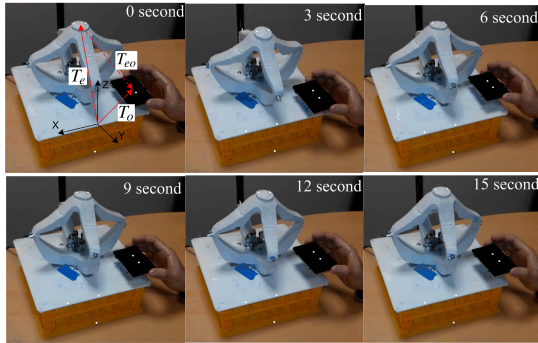


Fig. 6. Soft robot tracks a moving object which lies on the middle of the two markers placed on the black plane. The controlled point is the end-effector of the robot.

control method is affected by the inaccuracy of models. If the simulation model is highly imprecise, for example to deal with the dynamic problem with the controller based on quasi-static model, the control performance maybe unsatisfactory and even unstable. So a more complex control method is needed to increase the robustness and adaptivity of the controller which is the future work in the field of soft robot control.

REFERENCES

- [1] Daniela Rus and Michael T. Tolley. Design, fabrication and control of soft robots. *Nature*, 521(7553):467–475, 2015.
- [2] Robert J. Webster and Bryan A. Jones. Design and kinematic modeling of constant curvature continuum robots: A review. *The International Journal of Robotics Research*, 2010.
- [3] Hesheng Wang, Weidong Chen, Xiaojin Yu, Tao Deng, Xiaozhou Wang, and Rolf Pfeifer. Visual servo control of cable-driven soft robotic manipulator. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 57–62. IEEE, 2013.
- [4] Dinesh K. Pai. Strands: Interactive simulation of thin solids using cosserat models. In *Computer Graphics Forum*, volume 21, pages 347–352. Wiley Online Library, 2002.
- [5] Deepak Trivedi, Amir Lotfi, and Christopher D. Rahn. Geometrically exact dynamic models for soft robotic manipulators. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pages 1497–1502. IEEE, 2007.
- [6] Federico Renda, Michele Giorelli, Marcello Calisti, Matteo Cianchetti, and Cecilia Laschi. Dynamic model of a multibending soft robot arm driven by cables. *Robotics, IEEE Transactions on*, 30(5):1109–1122, 2014.
- [7] Panagiotis Polygerinos, Zheng Wang, Johannes T. B. Overvelde, Kevin C. Galloway, Robert J. Wood, Katia Bertoldi, and Conor J. Walsh. Modeling of soft fiber-reinforced bending actuators. *Robotics, IEEE Transactions on*, 31(3):778–789, 2015.
- [8] Christian Duriez. Control of elastic soft robots based on real-time finite element method. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 3982–3987. IEEE, 2013.
- [9] Eulalie Coevoet, Nick Reynaert, Eric Lartigau, Luis Schiappacasse, Jeremie Dequidt, and Christian Duriez. Introducing interactive inverse FEM simulation and its application for adaptive radiotherapy. In *Medical Image Computing and Computer-Assisted InterventionMICCAI 2014*, pages 81–88. Springer, 2014.
- [10] Frederick Largilliere, Valerian Verona, Eulalie Coevoet, Mario Sanz-Lopez, Jeremie Dequidt, and Christian Duriez. Real-time control of soft-robots using asynchronous finite element modeling. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 2550–2555. IEEE, 2015.
- [11] Julien Bosman, Thor Morales Bieze, Othman Lakhal, Mario Sanz, Rochdi Merzouki, and Christian Duriez. Domain decomposition approach for FEM quasistatic modeling and control of Continuum Robots with rigid vertebrae. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 4373–4378. IEEE, 2015.
- [12] Vishesh Vikas, Piyush Grover, and Barry Trimmer. Model-free control framework for multi-limb soft robots. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1111–1116. IEEE, 2015.
- [13] Dragan Nesic and Andrew R. Teel. Sampled-data control of nonlinear systems: an overview of recent results. In *Perspectives in robust control*, pages 221–239. Springer, 2001.
- [14] Guillaume Saupin, Christian Duriez, Stephane Cotin, and Laurent Grisoni. Efficient contact modeling using compliance warping. In *computer graphics international*, 2008.
- [15] Hadrien Courtecuisse, Jeremie Allard, Christian Duriez, and Stephane Cotin. Preconditioner-based contact response and application to cataract surgery. In *Medical Image Computing and Computer-Assisted Intervention-MICCAI 2011*, pages 315–322. Springer, 2011.
- [16] Eulalie Coevoet, Nick Reynaert, Eric Lartigau, Luis Schiappacasse, Jeremie Dequidt, and Christian Duriez. Registration by interactive inverse simulation: application for adaptive radiotherapy. *International journal of computer assisted radiology and surgery*, 10(8):1193–1200, 2015.